SELF-SUPERVISED LEARNING OF APPLIANCE USAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning home appliance usage is important for understanding people's activities and optimizing energy consumption. The problem is modeled as an event detection task, where the objective is to learn when a user turns an appliance on, and which appliance it is (microwave, hair dryer, etc.). Ideally, we would like to solve the problem in an unsupervised way so that the method can be applied to new homes and new appliances without any labels. To this end, we introduce a new deep learning model that takes input from two home sensors: 1) a smart electricity meter that outputs the total energy consumed by the home as a function of time, and 2) a motion sensor that outputs the locations of the residents over time. The model learns the distribution of the residents' locations conditioned on the home energy signal. We show that this cross-modal prediction task allows us to detect when a particular appliance is used, and the location of the appliance in the home, all in a self-supervised manner, without any labeled data.

1 INTRODUCTION

Learning home appliance usage patterns is useful for understanding user habits and optimizing electricity consumption. For example, knowing when a person uses their microwave, fridge, oven, coffee machine or dish washer provides information about their eating patterns. Similarly, understanding when they use their TV, air-conditioner, or washing machine provides knowledge of their behavior and habits. Such information may be used to encourage energy saving by optimizing appliance usage, to track the wellbeing of elderly living alone (Armel et al., 2013; Donini et al., 2013; Debes et al., 2016), or to provide users with behavioral analytics (Zhou & Yang, 2016; Zipperer et al., 2013). This data is also useful for various businesses such as home insurance companies interested in assessing accident risks and utility companies interested in optimizing energy efficiency (Armel et al., 2013).

The problem can be modeled as event detection – i.e., given the total energy consumed by the house as a function of time, we want to detect when various appliances are turned on. Past work has looked at analyzing the energy signal from the home utility meter to detect when certain appliances are on.[1] Most solutions, however, assume that the energy pattern for each appliance is unique and known, and use this knowledge to create labeled data for their supervised models. (Kolter et al., 2010; Zhong et al., 2014; 2015; Kelly & Knottenbelt, 2015; Zhang et al., 2018; Bonfigli et al., 2018). Unfortunately, such solutions do not generalize well because the energy pattern of an appliance depends on its brand and can differ from one home to another (Kelly & Knottenbelt, 2015; Bonfigli et al., 2018).[2] The literature also contains some unsupervised methods, but they typically have limited accuracy (Kim et al., 2011; Kolter & Jaakkola, 2012; Johnson & Willsky, 2013; Parson et al., 2014; Wytock & Kolter, 2014; Zhao et al., 2016; Lange & Berges, 2018).

Unsupervised event detection in a data stream is intrinsically challenging because we do not know what patterns to look for. In our task, not only may appliance energy patterns be unknown, but also the energy signal may include many background events unrelated to appliance activation, such as the fridge or HVAC power cycling events.

One way to address this challenge is to consider the self-supervised paradigm. If a different stream of data also observes the events of interest, we can use this second modality to provide self-supervising signals for event detection. To that end, we leverage the availability of new fine-resolution motion sensors which track the locations of people at home (Adib et al., 2015; Joshi et al., 2015; Li et al., 2016; Ghourchian et al., 2017; Hsu et al., 2017). Such sensors operate as a consumer radar, providing decimeter-level location accuracy. They do not require people to wear sensors on their bodies, can operate through walls, and track people's locations in different rooms.

[1] The utility meter outputs the sum of the energy of all active appliances in a house as a function of time.

[2] For example, a Samsung dishwasher may have a different energy pattern from that of a Kenmore dishwasher.

These location sensors indirectly observe the events of interest. Specifically, they capture the change in user locations as they reach out to an appliance to set it up or turn it on (put food in a microwave and turn it on). Hence, the output of such sensors can provide a second modality for self-supervision.

But how should one design the model? We cannot directly use location as a label for appliance activation events. People can be next to an appliance but neither activate it nor interact with it. We also cannot use the two modalities to learn a joint representation of the event in a shared space. This is because location and energy are unrelated most of the time and become related only when the event of interest occurs. Furthermore, there are typically multiple residents in the home, making it hard to tell which of them interacted with the appliance.

Our model is based on cross-modal prediction. We train a neural network that, given the home energy at a particular time, predicts the location of the home residents. Our intuition is that appliance activation events have highly predictable locations, typically the location of the appliance. In contrast, background energy events (power cycling of the fridge) do not lead to predictable locations. Thus, our model uses this learned predictability along with the associated location and energy representation to cluster the events in the energy stream. In addition, we formulate a mixture distribution to disentangle irrelevant location information from other residents in the home. Interestingly, our model not only learns when each appliance is activated but also discovers the location of that appliance in the home, all without any labeled data.

We summarize the contributions of this paper as follows:

- The paper introduces a new method for self-supervised event detection from weakly related data streams. The method combines neural cross-modal prediction with custom clustering based on the learned predictability and representation. We apply it to the task of detecting appliance usage events using unlabeled data from two sensors in the home: the energy meter, and a location sensor.
- To evaluate our design, we have created the first dataset with concurrent streams of home energy and location data, collected from 4 homes over a period of 7 months. For each home, data was collected for 2 to 4 months. Ground truth measurements are provided via smart plugs connected directly to each appliance.
- Compared against past work on unsupervised learning of appliance usage and a new baseline that leverages the two modalities, our method achieves significant improvements of 66.7% and 48.8% respectively for the average detection F1 score.

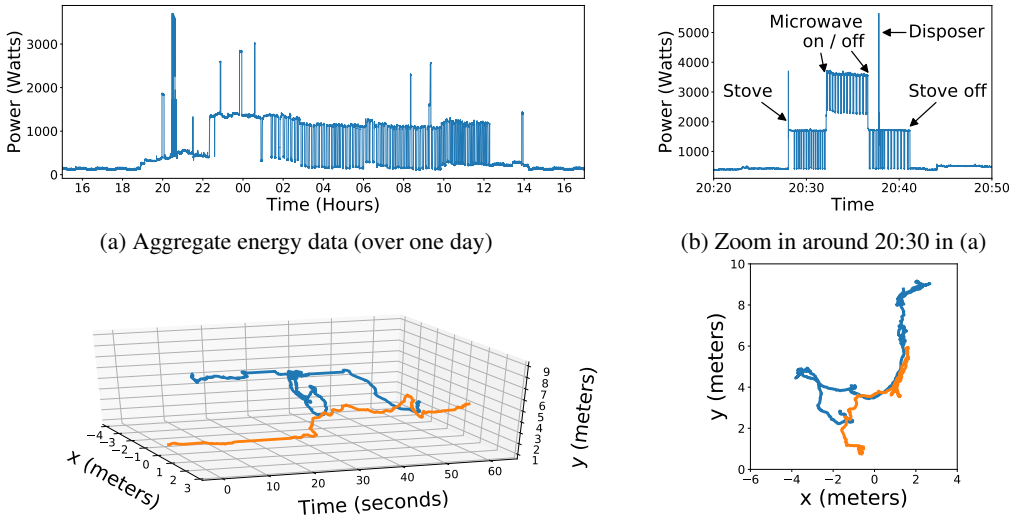
We will release our code and dataset to encourage future work on multi-modal models for understanding appliance usage patterns and the underlying user behavior.

2 Related Work

Energy disaggregation Our work is related to past work on energy disaggregation, which refers to the problem of separating appliance-level energy from a home's total (or aggregate) energy signal. Past work in this domain can be broadly classified into two categories: supervised and unsupervised.

Supervised methods assume that the power signatures of individual appliances are available. They use data from individual appliances to obtain models for each appliance power signature, and then use those models to detect appliance events from the aggregate energy signal. Early work learns sparse codes for different appliances (Kolter et al., 2010) or uses a Factorial HMM (FHMM) (Ghahramani & Jordan, 1996) to model each appliance as an HMM (Zhong et al., 2014; 2015). More recently, neural networks have been used to model appliances (Kelly & Knottenbelt, 2015; Zhang et al., 2018; Bonfigli et al., 2018), where extracting appliance-level energy is formulated as a de-noising problem. However, supervised solutions do not generalize well to new homes (Kelly & Knottenbelt, 2015; Bonfigli et al., 2018). This is because two appliances of the same type (e.g. coffee machine) in different homes are often manufactured by different brands, and thus have different power signatures.

Unsupervised methods do not assume prior knowledge of appliance signatures; they attempt to learn those signatures from the aggregate energy signal. Early approaches use variants of FHMM, and learn appliance HMMs with Expectation-Maximization (Kim et al., 2011), approximate footprint extraction procedures (Kolter & Jaakkola, 2012), or use expert knowledge to configure prior parameters (Johnson & Willsky, 2013; Parson et al., 2014). Some papers propose using contextual information (such as temperature, hour of the day, and day of the week) (Wytock & Kolter, 2014), or use event-based signal processing methods to cluster appliances (Zhao et al., 2016). More recently, Lange & Berges (2018) proposed using a recurrent neural network as the variational distribution in learning the FHMM. In



(a) Aggregate energy data (over one day)

(b) Zoom in around 20:30 in (a)

(c) Location data for two people over one minute

(d) Location data for two people (top-down view)

Figure 1: Aggregate energy signal and people's indoor location data

contrast, our work leverages people's location data as a self-supervising signal. We cluster appliance events through learning the relation between energy events and people's locations, and also learn appliance locations as a by-product.

Passive location sensing Motivated by new in-home applications and continuous health monitoring, recent years have witnessed an increasing number of indoor location sensing systems (Adib et al., 2015; Joshi et al., 2015; Li et al., 2016; Ghourchian et al., 2017). They infer people's locations passively by analyzing how people change the surrounding radio signals (e.g. WiFi) and do not require people to wear any sensors. These sensors have been used for various applications including activity recognition (Wang et al., 2014; 2015), sleep monitoring (Zhao et al., 2017), mobility tracking (Hsu et al., 2017), and health monitoring (Kaltiokallio et al., 2012). In our work, we leverage the availability of such sensors to introduce location data as an additional data modality for learning appliance usage patterns.

Self-supervised multi-modal learning Our work is related to a growing body of work on multi-modal learning. Most approaches learn to encode the multi-modal data into a shared space (Harwath et al., 2018; Owens & Efros, 2018; Zhao et al., 2018). In contrast, since our two modalities are mostly unrelated and become related only when an activation event happens, we learn to predict one modality conditioned on the other. Our work is also related to cross-modal prediction (Krishna et al., 2017; Owens et al., 2016; Zhang et al., 2017) but differs from it in an essential way. Past work on cross-modal prediction typically uses the prediction as the target outcome (e.g. output text for video captioning). In contrast, our objective is to discover the hidden appliance activation events. Thus, we design our method to leverage the learned predictability and cross-modal mapping for clustering activation events. Furthermore, we introduce a mixture prediction design to disentangle unrelated information in our predicted modality (location measurements unrelated to energy events).

3 Problem Formulation

Our goal is to learn appliance activation events in an unsupervised way, using two input streams: home aggregate energy and residents' location data. Figure 1 shows the two data modalities. We describe each of them formally and define appliance "events" below.

Aggregate energy signal A household's total energy consumption is measured by a utility meter regularly. This measures the sum of energy consumption from all appliances at each point in time. We denote the aggregate energy signal by $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$, where $y_t \in \mathbb{R}_+$. Suppose there are a total of K appliances in a home, and each appliance's energy signal is denoted by $\boldsymbol{x}_k = (x_{1,k}, x_{2,k}, \ldots, x_{T,k})$, where $x_{t,k} \in \mathbb{R}_+$. Only the aggregate energy signal is observed, $y_t = \sum_{k=1}^{K} x_{t,k} + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ is the background noise.

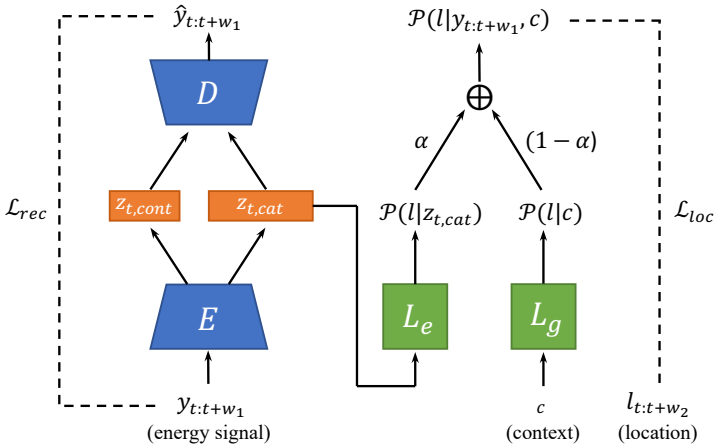


Figure 2: Model architecture. The model learns to encode energy signals into event vectors while learning to predict the concurrent location data. The location predictor L_e is conditioned on the energy event $z_{t,cat}$, and L_g is conditioned on context features. The decoder D takes event vectors and learns to reconstruct the original energy signal.

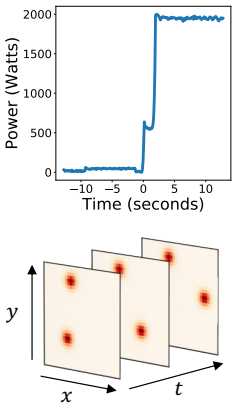


Figure 3: Total energy signal (top) and location information (bottom) as seen by the model

Figure 1a shows one day of an aggregate energy signal. The base power level shifts constantly throughout a day, depending on the background load (e.g. ceiling lights). Added on top of the base level are the various appliance events. Figure 1b zooms in around 20:30, and shows examples of those events. The stove was turned on around 20:28, and its power continues to cycle between a few levels. While the stove was on, the microwave was also turned on and ran for a few minutes, and the garbage disposer was turned on shortly.

Indoor location data We use a single location sensor similar to that in Hsu et al. (2017) to measure people's indoor locations passively. The sensor sends out radio signals and analyzes the reflections to localize multiple people. Similarly to a regular WiFi router, the sensor has a limited coverage area of up to 40 feet. Suppose there are P_t people in the coverage area at time t. The location data is denoted by $\boldsymbol{l}_t = (l_{t,1}, l_{t,2}, \ldots, l_{t,P_t})$, where $l_{t,p} \in \mathbb{R}^2$ is the x-y location at time t of person p. We can represent the location data over multiple time frames as $\boldsymbol{l}_{1:T} = (\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_T)$. Figure 1c shows one minute of location data from two people, and Figure 1d shows the data from a top-down view.

Appliance activation events When an appliance is turned on, it causes a jump in energy consumption, i.e. a leading edge in the energy signal, as shown in Figure 1b. We call such a pattern an appliance activation event. On the other hand, when an appliance changes its internal state, it can also cause a change in the energy signal as shown in the same figure. We call such a pattern a background event. We are interested in discovering activation events to learn appliance usage patterns. Thus, for each jump in the aggregate signal, we take a time window (default is 25 seconds) centered around that jump, and analyze it to detect whether it is an activation event and which appliance it corresponds to.

4 Model

Our model operates on time windows (25 seconds) centered around jumps in aggregate energy signal, and the corresponding time windows of location data. The model aims to detect appliance activation events by finding windows with highly predictable user locations conditioned on the energy signal.

Figure 2 shows our model. The idea underlying our model is to first learn a representation of appliance event windows that separate the appliance type, $z_{t,cat}$, from the shape of the energy signal, $z_{t,cont}$ in the figure. This is achieved through the appliance energy encoder E. We can then use the appliance type to predict the location data through the location predictor L_e, which is conditioned on $z_{t,cat}$. Since people's locations have information unrelated to appliance events, the total location predictor is a mixture of L_e and a second module L_g which captures event-independent location information. Below, we describe these modules in detail.

Appliance Energy Encoder Given a window of aggregate energy signal $\boldsymbol{y}_{t:t+w_1} = (y_t, y_{t+1}, \ldots, y_{t+w_1})$,[3] the encoder E encodes the series into an event vector $\boldsymbol{z}_t$. We break the

[3]We remove the base power level in each window by subtracting the minimum in the window.

event vector into two parts: a categorical vector $\boldsymbol{z}_{t,cat}$ and a continuous vector $\boldsymbol{z}_{t,cont}$. We aim to capture the appliance type with $\boldsymbol{z}_{t,cat}$ (e.g. microwave vs. dishwasher), and use $\boldsymbol{z}_{t,cont}$ to capture the variability within an appliance. A softmax layer is applied to $\boldsymbol{z}_{t,cat}$ to ensure that it is a valid distribution over appliance types. E is parametrized using convolution layers, with one fully connected layer to produce $\boldsymbol{z}_{t,cont}$ and another for $\boldsymbol{z}_{t,cat}$. We denote by $\boldsymbol{\theta}_E$ the parameters of the encoder.

Location predictors We try to predict the location data conditioned on the appliance event, i.e. we predict a window of locations $\boldsymbol{l} = \boldsymbol{l}_{t:t+w_2} = (\boldsymbol{l}_t, \boldsymbol{l}_{t+1}, \ldots, \boldsymbol{l}_{t+w_2})$ centered around the appliance event. We handle multiple people's locations with a mixture model. Specifically, we use L_e to predict locations related to energy events and L_g to handle other locations. The final prediction is a mixture of predictions from L_e and L_g:

$$p_{\boldsymbol{\theta}_L}(\boldsymbol{l}|\boldsymbol{y}_{t:t+w_1}, \boldsymbol{c}) = \alpha * p_{\boldsymbol{\theta}_{L_e}}(\boldsymbol{l}|\boldsymbol{z}_{t,cat}) + (1-\alpha) * p_{\boldsymbol{\theta}_{L_g}}(\boldsymbol{l}|\boldsymbol{c}),$$

where $p_{\boldsymbol{\theta}_{L_e}}(\cdot)$ is parametrized by L_e with parameters $\boldsymbol{\theta}_{L_e}$, $p_{\boldsymbol{\theta}_{L_g}}(\cdot)$ is parametrized by L_g with parameters $\boldsymbol{\theta}_{L_g}$, $\boldsymbol{\theta}_L = \left\{\boldsymbol{\theta}_{L_e}, \boldsymbol{\theta}_{L_g}\right\}$, and $\boldsymbol{c}$ includes context features. We use the number of people in the window (reported by the location sensor), the time of day, and the day of the week as the context features. The weight α depends on the number of people in the current window $\alpha = 1/P_t$.

To represent location data, we blur each location measurement with a Gaussian kernel on the x-y plane to create an image, and process the window of locations $\boldsymbol{l}_{t:t+w_2}$ into frames of images. We reuse the notation $\boldsymbol{l}$ to represent frames of location images $\in \mathbb{R}^{|X|\times|Y|\times|T|}$, where $|X|$, $|Y|$, $|T|$ are the number of discretized points on the x-y and time dimensions. By presenting location data as images, we also remove the variable P_t while handling a variable number of people in each frame.

We choose $p_{\boldsymbol{\theta}_{L_e}}(\boldsymbol{l}|\boldsymbol{z}_{t,cat})$ to be a multivariate Gaussian with a diagonal covariance structure: $p_{\boldsymbol{\theta}_{L_e}}(\boldsymbol{l}|\boldsymbol{z}_{t,cat}) = \mathcal{N}(\boldsymbol{l}; \boldsymbol{\mu}_e, \boldsymbol{\Sigma}) = \prod_{x,y,t} \mathcal{N}(l_{x,y,t}; \mu_{x,y,t}, \sigma^2_{x,y,t})$ where $\boldsymbol{\mu}_e = L_e(\boldsymbol{z}_{t,cat}; \boldsymbol{\theta}_{L_e})$ and choose $\sigma_{x,y,t}$ to be a constant. We use 3D deconvolution networks to model L_e, which takes $\boldsymbol{z}_{t,cat}$ as input and outputs the means of the location distributions. We model $p_{\boldsymbol{\theta}_{L_g}}(\cdot)$ and L_g in a similar way. The network implementation details are discussed in Appendix 8.4.

During training, given a window of data $(\boldsymbol{l}^{(i)}, \boldsymbol{y}^{(i)}_{t:t+w_1}, \boldsymbol{c}^{(i)})$, we minimize the negative log likelihood of the mixture distribution in predicting the locations:

$$\mathcal{L}_{loc}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_L; \boldsymbol{l}^{(i)}) = -\log\ p_{\boldsymbol{\theta}_L}(\boldsymbol{l}^{(i)}|\boldsymbol{y}^{(i)}_{t:t+w_1}, \boldsymbol{c}^{(i)}).$$

Note that as the gradient flows through $\boldsymbol{z}_{t,cat}$, the likelihood is a function of both $\boldsymbol{\theta}_E$ and $\boldsymbol{\theta}_L$. Hence, the encoder E also learns to encode the energy series based on the concurrent location data.

Energy Decoder The decoder D takes both $\boldsymbol{z}_{t,cat}$ and $\boldsymbol{z}_{t,cont}$ and learns to reconstruct the original input energy series by predicting $\hat{\boldsymbol{y}}_{t:t+w_1}$. We minimize the reconstruction loss during training:

$$\mathcal{L}_{rec}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_D) = ||\hat{\boldsymbol{y}}_{t:t+w_1} - \boldsymbol{y}_{t:t+w_1}||_2$$

The reconstruction loss encourages the encoder E to produce good initial vectors for L_e to predict locations. At the same time, it serves as a regularizer to prevent encoder E from generating meaningless vectors by overfitting location predictions.

Training We train all components to jointly optimize the location predictions and energy reconstruction. We minimize the total loss: $\mathcal{L}_{total} = \mathcal{L}_{loc} + \lambda * \mathcal{L}_{rec}$ where λ is a parameter to balance the two terms. [4] The network implementation and training details are discussed in Appendix 8.4.

4.1 Clustering Appliance Events with Cross-Modal Predictions

Once the model is trained, we obtain for each window of energy data its appliance event vector $\boldsymbol{z}_{t,cat}$ and its cross-modal location prediction $p_{\boldsymbol{\theta}_{L_e}}(\cdot|\boldsymbol{z}_{t,cat})$. Next, we use these two vectors for clustering. We design a density-based clustering algorithm leveraging the cross-modal relation we learned. Our intuition is that activation events for the same appliance will cluster together since they have the same appliance type and the same location. We omit the cat notation below for brevity.

It is typically difficult to cluster in a space learned by a neural encoder because the transformation is highly non-linear and the distance metric is not well-defined. We circumvent this problem by

[4] We choose λ to be 0.1 in our experiments to put more emphasis on the location prediction.

associating the encoded space with a Euclidean space, in which we can easily measure distance. Specifically, for two event vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, we can measure their distance in the location space using $p_{\boldsymbol{\theta}_{L_e}}(\cdot|\boldsymbol{z}_1)$ and $p_{\boldsymbol{\theta}_{L_e}}(\cdot|\boldsymbol{z}_2)$.

The location prediction $p_{\boldsymbol{\theta}_{L_e}}(\cdot|\boldsymbol{z}_i)$ represents the likelihood of observing any location $l_{x,y,t}$ around the time of the appliance event. We found that for events related to human activities (e.g., turning on a kettle or microwave), $p_{\boldsymbol{\theta}_{L_e}}(\cdot|\boldsymbol{z}_i)$ shows a peak value at the location of the appliance in the x-y space at the time when a person interacted with the appliance. For events not related to human activities (e.g. fridge cycles or random background events), $p_{\boldsymbol{\theta}_{L_e}}(\cdot|\boldsymbol{z}_i)$ has low values and is diffused.

We define the location predictability score (or the confidence of location prediction) as $s(\boldsymbol{z}_i) = \max_{x,y,t}\ p_{\boldsymbol{\theta}_{L_e}}(l_{x,y,t}|\boldsymbol{z}_i)$, and the location distance D_{loc} between two events as: $D_{\text{loc}}(\boldsymbol{z}_1, \boldsymbol{z}_2) = ||(x_1^* - x_2^*, y_1^* - y_2^*)||_2$, where $(x_i^*, y_i^*, t_i^*) = \arg\max_{x,y,t}\ p_{\boldsymbol{\theta}_{L_e}}(l_{x,y,t}|\boldsymbol{z}_i)$. Similarly, the neighborhood distance D_{nb} between two events is defined as $D_{\text{nb}}(\boldsymbol{z}_1, \boldsymbol{z}_2) = ||\boldsymbol{z}_1 - \boldsymbol{z}_2||_2$.

Our clustering algorithm starts with a $\boldsymbol{z}_i$ with high predictability score $s(\boldsymbol{z}_i)$. It expands the cluster around $\boldsymbol{z}_i$'s local neighborhood in the $\boldsymbol{z}$ space. It stops expanding if a neighbor's location distance D_{loc} is too far from the cluster center. If all neighbors of the current cluster are visited and none has a small enough D_{loc}, we start a new cluster from another event with high predictability score. The algorithm is described formally in Algorithm 1. We discuss the choice of parameters in Appendix 8.5.

Algorithm 1 Clustering energy events with the learned cross-modal relations

Input: $\{\boldsymbol{z}_i\}$ and $s(\cdot)$: event vectors and their location predictability scores
η_s, $\eta_{D_{\text{loc}}}$, η_z: thresholds for predictability score, location distance, neighborhood distance
N_{min}: the minimum number of samples to form a valid cluster
Output: Clusters of appliance activation events that are associated with a consistent location

```
Z ← {z_i | s(z_i) > η_s}
procedure EL-SCAN(η_s, η_{D_loc}, η_z, N_min)
    while Z ≠ ∅ do
        z_seed = arg max_Z s(z_i)
        cluster_k ← {z_seed}
        ExpandCluster(k, z_seed, η_s, η_{D_loc}, η_z)
    end while
    return clusters with at least N_min examples
end procedure

function EXPANDCLUSTER(k, z, η_s, η_{D_loc}, η_z)
    z_{u_k} ← compute current cluster center
    Z_nb ← {z_i ∈ Z | D_nb(z_i, z) < η_z and D_loc(z_{u_k}, z_i) < η_{D_loc}}    ▷ Find valid neighbors
    Z ← Z \ Z_nb
    cluster_k ← cluster_k ∪ Z_nb
    Repeat ExpandCluster(.) for all z_i in Z_nb
end function
```

5 DATASET

We collected concurrent streams of aggregate energy signal and location data from 4 homes over 7 months. We use this dataset for our evaluation. To obtain ground truth labels of appliance events, we deployed programmable smart plugs on the power outlet associated with each appliance. Since not all appliances can be measured by a smart plug (some appliances do not connect to a power outlet), we also developed a labeling tool for manual labeling. The tool allows labelers to label appliance events from the aggregate energy signal, with the help of smart plug data and information collected from the home residents. The choice of sensors and their sampling rates are detailed in Appendix 8.1.

6 RESULTS

We evaluate our model and clustering algorithm on unsupervised appliance activation event detection and their learned appliance locations.

6.1 UNSUPERVISED APPLIANCE EVENT DETECTION

For appliance event detection, we compare with four baselines. Our method and two baselines have access to location information. *EL-Kmeans* takes both energy and location data as input and directly

clusters them using k-means (Arthur & Vassilvitskii, 2007).[5] *E-only-Kmeans* clusters only the energy signal with k-means. Methods with location information pre-filter the events and discard events without any location data, as they are unlikely to be activation events. The other two baselines only take the total energy signal as input: *AFAMAP* (Kolter & Jaakkola, 2012) uses factorial HMM, and *VarBOLT* (Lange & Berges, 2018) uses a recurrent neural network to model aggregate appliance signals. We use publicly available implementations for these methods (implementation, a;b).

We use the same hyper-parameters for the network architecture, training, and clustering algorithm across all homes. As our clustering algorithm is non-parametric, we choose the same number of clusters that it discovers for other methods if possible. For VarBOLT, we report results using 10 clusters, since the training time grows exponentially with the number of clusters and training with more clusters is prohibitively slow. As in past unsupervised work, we report the detection F1 scores based on the best cluster assignments with the ground truth appliances.

Table 1 shows that our method has an average detection F1 score of 73.2%, outperforming other baselines ranging from 4.1% to 24.4%. As reported by Bonfigli et al. (2018), AFAMAP performs better when appliance-level data is available for training the HMMs. In the unsupervised setting, however, its footprint extraction procedure does not always produce meaningful HMMs for individual appliances (Bonfigli et al., 2018; Beckel et al., 2014), causing degraded performance. VarBOLT's training objective focuses on explaining the total amount of energy in a home. Thus, it often uses multiple components (clusters) to model appliances that are on for a long period (e.g. fridge, heater, and dryer/washer). These types of appliances generate many background events, making the algorithm focus less on activation events of other appliances.

Comparing our method with baselines that also have location information (E-only-Kmeans and EL-Kmeans), our approach still outperforms them significantly. E-only-Kmeans performs better than AFAMAP and VarBOLT, showing that the presence of location data is highly related to activation events. However, naively using the location data for clustering does not improve the results, as EL-Kmeans performs similarly to E-only-Kmeans. This is because not all location data is related to appliance events and vice versa. Our approach "cleans up" the data by learning the relation between the two modalities and discovers clusters with strong cross-modal predictability.

Table 2 shows a break down of our results for different appliances.

Table 1: Unsupervised Appliance Event Detection. Averaged F1 scores (%) of all appliances.

		Methods w/ location information			Methods w/o location	
	$N_{\text{appliances}}$	**Ours**	EL-Kmeans	E-only-Kmeans	AFAMAP	VarBOLT
Home 1	8	**82.3**	34.8	23.7	6.1	4.7
Home 2	8	**69.7**	21.6	21.7	5.7	3.9
Home 3	6	**76.2**	17.3	22.2	5.2	3.4
Home 4	6	**64.4**	23.9	27.1	9.1	4.5
Average	-	**73.2**	24.4	23.7	6.5	4.1

6.2 Ablation Study

We perform an ablation study to show our results are contributed by all components in our method. As shown in Table 3, we compare our clustering algorithm (Method 1) with a different algorithm that concatenates the learned multi-modal embeddings ($\boldsymbol{z}_{t,cat}$ and $p_{\boldsymbol{\theta}_{L_e}}(\cdot|\boldsymbol{z}_{t,cat})$) and directly clusters them with k-means (Method 2). Our clustering algorithm is more effective than directly clustering the multi-modal embeddings, providing an improvement of 13.2% in the average F1 score. This is because our clustering algorithm treats the two modalities differently. For location predictions, we can leverage our understanding of physical distance to set cluster boundaries. For the energy embedding, since it is a non-linear mapping with no clear distance metric, our algorithm iteratively groups together events in the embedding neighborhood that have approximately the same locations.

Apart from our clustering algorithm, we evaluate the benefits of our mixture component L_g by experimenting with removing L_g from the model, which reduces the F1 score by 4.1% (Method 1 vs 3). This shows the importance of having L_g extract background motion to allow the location predictor, L_e, to focus on modeling the person who interacts with the appliance.

[5]For each window of data, *EL-Kmeans* concatenates the energy signal, the frames of location images (flattened as a 1-d vector), and the context vector to create the feature vectors clustered by k-means.

Table 2: Unsupervised Appliance Event Detection. Our method's F1 score (%) for each appliance.

	Home 1	Home 2	Home 3	Home 4
Kettle	91.9	-	-	98.6
Hair dryer	88.0 / 98.3	-	-	1.1
Coffee machine	96.1	75.8	90.4	-
Microwave	81.9	82.1	88.1	96.7
Stove-activation	90.6	88.1	-	92.7
Disposer	62.5	78.5	53.1	-
Toaster	-	48.6	71.1	-
Blender	-	9.4	-	-
Door / leaving	-	96.1	83.4	-
Iron	-	-	71.1	-
Rice Cooker	-	-	-	1.1
Others	49.4	76.7	-	96.3
Average	82.3	69.7	76.2	64.4

Table 3: Ablation study.

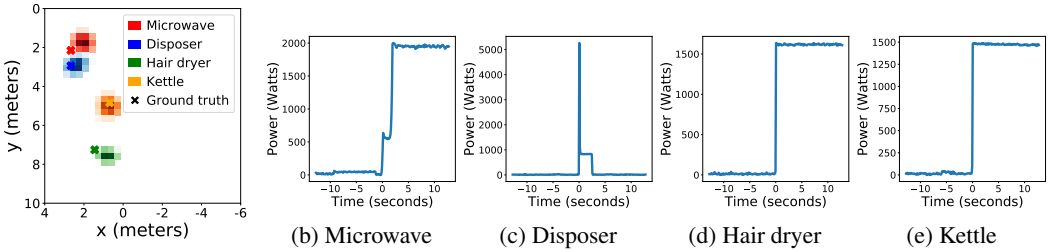
	Methods	Avg. F1 (%)
1	**Ours**	**73.2**
2	Learned embeddings + K-means	60.0
3	Remove L_g	69.1
4	Remove L_e & L_g	32.9

We also consider removing both L_g and L_e, and clustering the input based only on the energy embedding $\boldsymbol{z}_{t,cat}$ since there is no learned location predictions. The results shown under Method 4 demonstrate the importance of the location embedding generated by the combination of L_g and L_e.

6.3 Learned Appliance Locations

Our model also learns the locations where people interacted with appliances, which are typically close to the appliances' physical locations (we discuss remotely activated appliances in Appendix 8.6). For each appliance event, we take the location predicted by L_e with the highest predictability score, and compare that with the ground truth appliance location measured by a laser meter. The average location prediction error is 0.77 meters with a standard deviation of 0.48 meters across homes. Figure 4a shows the location predictions and their ground truth of several appliance events in Home 1. The corresponding energy signals are shown in Figure 4b - Figure 4e.

The location information helps disambiguate appliances with similar energy signals. For example, although the hair dryer and kettle (Figure 4d and Figure 4e) have very similar energy signatures, their different locations (green and orange in Figure 4a) guide the model to encode their events differently.



(a) Learned locations (b) Microwave (c) Disposer (d) Hair dryer (e) Kettle

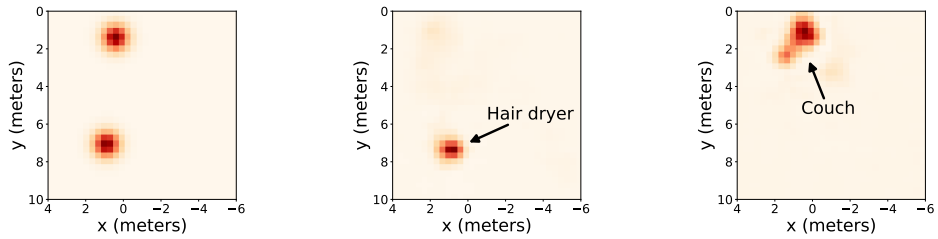
Figure 4: Energy signals of discovered activation events and their learned locations from L_e.

6.4 Location Predictions of L_e vs L_g

We visualize the location predictions from the event-related predictor L_e and the event-independent predictor L_g to illustrate how they handle scenarios with multiple people. Figure 5 shows how the mixture design handles the two types of locations. Since L_e is conditioned on energy events, it naturally learns to predict locations related to appliance events. In this case, the location of the hair dryer is predicted by L_e (Figure 5b). On the other hand, L_g predicts the typical locations people tend to stay (e.g., the couch in Figure 5c) based on the context. Having L_g to explain the other locations helps L_e focus on learning the event-related locations.

6.5 Contextual Location Information and Cluster Visualizations

In Appendix 8.2, we discuss emerging contextual relations between indoor locations through learning cross-modal predictions. We also visualize the learned event vectors to shed light on the design rationales behind our clustering algorithm in Appendix 8.3.



(a) Observed locations (two people) (b) L_e's prediction (event-related) (c) L_g's prediction (other locations)

Figure 5: Observed locations and predictions of L_e and L_g at a given time for a hair dryer event.

7 Conclusion

We introduced a self-supervised solution for learning appliance usage patterns in homes. We infer appliance usage by learning from data streams of two modalities: the total energy consumed by the home and the residents' location data. Our approach improves on unsupervised appliance event detection significantly, and learns appliance locations and usage patterns without any supervision. [6]

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Fadel Adib, Zachary Kabelac, and Dina Katabi. Multi-person localization via rf body reflections. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pp. 279–292, 2015.

K Carrie Armel, Abhay Gupta, Gireesh Shrimali, and Adrian Albert. Is disaggregation the holy grail of energy efficiency? the case of electricity. *Energy Policy*, 52:213–234, 2013.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Christian Beckel, Wilhelm Kleiminger, Romano Cicchetti, Thorsten Staake, and Silvia Santini. The eco data set and the performance of non-intrusive load monitoring algorithms. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, pp. 80–89. ACM, 2014.

Roberto Bonfigli, Andrea Felicetti, Emanuele Principi, Marco Fagiani, Stefano Squartini, and Francesco Piazza. Denoising autoencoders for non-intrusive load monitoring: improvements and comparative evaluation. *Energy and Buildings*, 158:1461–1474, 2018.

Christian Debes, Andreas Merentitis, Sergey Sukhanov, Maria Niessen, Nikolaos Frangiadakis, and Alexander Bauer. Monitoring activities of daily living in smart homes: Understanding human behavior. *IEEE Signal Processing Magazine*, 33(2):81–94, 2016.

Lorenzo Maria Donini, Eleonora Poggiogalle, Maria Piredda, Alessandro Pinto, Mario Barbagallo, Domenico Cucinotta, and Giuseppe Sergi. Anorexia and eating patterns in the elderly. *PloS one*, 8 (5):e63539, 2013.

emonPi. Open energy monitor `https://openenergymonitor.org/`.

Zoubin Ghahramani and Michael I Jordan. Factorial hidden markov models. In *Advances in Neural Information Processing Systems*, pp. 472–478, 1996.

Negar Ghourchian, Michel Allegue-Martinez, and Doina Precup. Real-time indoor localization in smart homes using semi-supervised learning. In *Twenty-Ninth IAAI Conference*, 2017.

[6] We discuss some limitations in Appendix 8.6.

David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 649–665, 2018.

Chen-Yu Hsu, Yuchen Liu, Zachary Kabelac, Rumen Hristov, Dina Katabi, and Christine Liu. Extracting gait velocity and stride length from surrounding radio signals. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 2116–2126. ACM, 2017.

AFAMAP implementation. `https://github.com/beckel/nilm-eval/tree/master/Matlab/algorithms/kolter_alg`, a.

Variational BOLT implementation. `https://github.com/INFERLab/varbolt`, b.

Matthew J Johnson and Alan S Willsky. Bayesian nonparametric hidden semi-markov models. *Journal of Machine Learning Research*, 14(Feb):673–701, 2013.

Kiran Joshi, Dinesh Bharadia, Manikanta Kotaru, and Sachin Katti. Wideo: fine-grained device-free motion tracing using rf backscatter. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pp. 189–204, 2015.

Ossi Kaltiokallio, Maurizio Bocca, and Neal Patwari. Follow@ grandma: Long-term device-free localization for residential monitoring. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, pp. 991–998. IEEE, 2012.

Jack Kelly and William Knottenbelt. Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, pp. 55–64. ACM, 2015.

Hyungsul Kim, Manish Marwah, Martin Arlitt, Geoff Lyon, and Jiawei Han. Unsupervised disaggregation of low frequency power measurements. In *Proceedings of the 2011 SIAM international conference on data mining*, pp. 747–758. SIAM, 2011.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

J Zico Kolter and Tommi Jaakkola. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Artificial intelligence and statistics*, pp. 1472–1482, 2012.

J Zico Kolter, Siddharth Batra, and Andrew Y Ng. Energy disaggregation via discriminative sparse coding. In *Advances in Neural Information Processing Systems*, pp. 1153–1161, 2010.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 706–715, 2017.

Henning Lange and Mario Berges. Variational bolt: approximate learning in factorial hidden markov models with application to energy disaggregation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Xiang Li, Shengjie Li, Daqing Zhang, Jie Xiong, Yasha Wang, and Hong Mei. Dynamic-music: accurate device-free indoor localization. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 196–207. ACM, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 631–648, 2018.

Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2405–2413, 2016.

Oliver Parson, Siddhartha Ghosh, Mark Weal, and Alex Rogers. An unsupervised training method for non-intrusive appliance load monitoring. *Artificial Intelligence*, 217:1–19, 2014.

TP-link. Tp-link smart plug hs110 `https://www.tp-link.com/uk/home-networking/smart-plug/hs110/`.

Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st annual international conference on mobile computing and networking*, pp. 65–76. ACM, 2015.

Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pp. 617–628. ACM, 2014.

Matt Wytock and J Zico Kolter. Contextually supervised source separation with application to energy disaggregation. In *Twenty-eighth AAAI conference on artificial intelligence*, 2014.

Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1058–1067, 2017.

Bochao Zhao, Lina Stankovic, and Vladimir Stankovic. On a training-less solution for non-intrusive appliance load monitoring using graph signal processing. *IEEE Access*, 4:1784–1799, 2016.

Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 570–586, 2018.

Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S Jaakkola, and Matt T Bianchi. Learning sleep stages from radio signals: A conditional adversarial architecture. In *International Conference on Machine Learning*, pp. 4100–4109, 2017.

Mingjun Zhong, Nigel Goddard, and Charles Sutton. Signal aggregate constraints in additive factorial hmms, with application to energy disaggregation. In *Advances in Neural Information Processing Systems*, pp. 3590–3598, 2014.

Mingjun Zhong, Nigel Goddard, and Charles Sutton. Latent bayesian melding for integrating individual and population models. In *Advances in neural information processing systems*, pp. 3618–3626, 2015.

Kaile Zhou and Shanlin Yang. Understanding household energy consumption behavior: The contribution of energy big data analytics. *Renewable and Sustainable Energy Reviews*, 56:810–819, 2016.

Adam Zipperer, Patricia A Aloise-Young, Siddharth Suryanarayanan, Robin Roche, Lieko Earle, Dane Christensen, Pablo Bauleo, and Daniel Zimmerle. Electric energy management in the smart home: Perspectives on enabling technologies and consumer behavior. *Proceedings of the IEEE*, 101(11):2397–2408, 2013.

8 APPENDIX

8.1 SENSORS DETAILS

In this section, we describe details of the sensors used in our dataset collection.

Aggregate energy signal For flexible data collection, we install a sensor (emonPi) at the main circuit breaker in each house as a proxy for the utility meter. We programmed the sensor to collect the raw aggregate energy signal at 1.2 kHz. We down-sampled the data to 10 Hz for our problem to emulate the achievable data rate from a utility meter hardware (Armel et al., 2013).

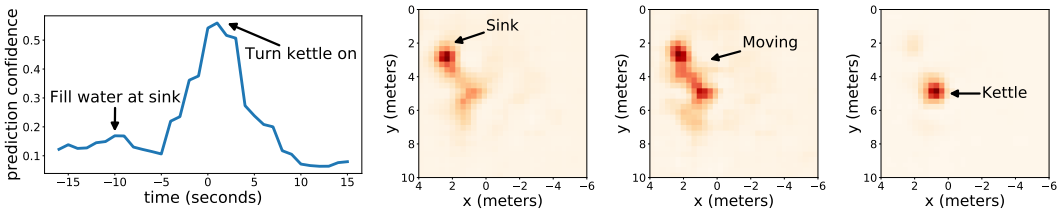
Location data The wireless location sensor is built on a design similar to Hsu et al. (2017). It is a single stand-alone sensor that hangs on the wall, and passively collects multiple people's locations with decimeter-level accuracy. We down-sampled the location streams to 1 Hz.

Appliance-level data (for ground truth labeling) We use TP-Link smart plugs (TP-link) with energy monitoring features for collecting appliance-level data. We wrote custom software using available APIs to collect appliance energy signals at 1 Hz. For appliances that cannot be connected to a smart plug, we asked the residents to write down appliance usage times to help with manual labeling.

8.2 CONTEXTUAL LOCATION INFORMATION VIA LEARNED APPLIANCE EVENTS

By analyzing the location predictions of L_e conditioned on different appliance events, we also discover interesting contextual relations between different indoor locations. Figure 6 visualizes the location predictions at different frames around a kettle event. We plot the per-frame location predictability score (or prediction confidence) over time in Figure 6a. The score peaks around $t = 0$s, the time of the event. This is because when people turn on a kettle, they may approach it from different locations, but the location when they push the button is consistent and can be predicted confidently. As a result, the prediction at $t = 0$s correctly shows the kettle's location (Figure 6d).

Interestingly, a smaller peak of predictability score shows at $t = -10$s in Figure 6a. If we look at the location prediction from $t = -10$s to $t = 0$s (Figure 6b - Figure 6d), we see how the prediction moves from the sink to the kettle [7]. This is because people often fill water at the sink before starting the kettle. Through learning the cross-modal relation, contextual information among locations also emerges as different appliance events are discovered.



(a) Prediction confidence over time (b) t = -10s (c) t = -6s (d) t = 0s

Figure 6: Visualizing location predictions at different times conditioned on a kettle event.

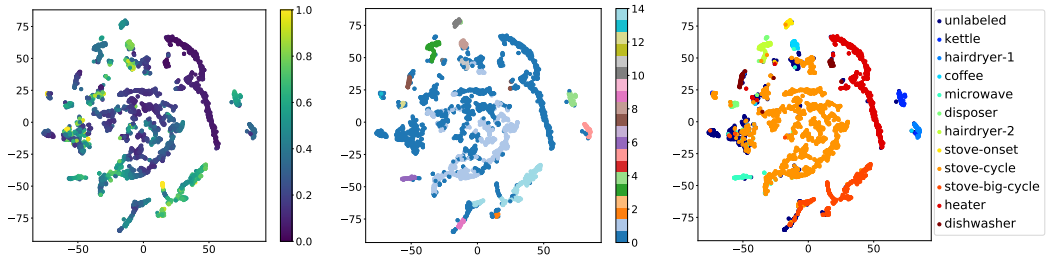
8.3 VISUALIZATION OF THE LEARNED EVENT VECTORS AND LOCATION PREDICTABILITY

To illustrate what the model learns and the design rationales behind our clustering algorithm, we visualize the space of the learned event vectors $\boldsymbol{z}_{t,cat}$ and their location predictability score $s(\boldsymbol{z})$. Figure 7 shows the t-SNE (Maaten & Hinton, 2008) visualization of the event vectors on a 2-dimensional space. We color coded the events with three metrics: location predictability scores (Figure 7a), cluster ID discovered by our algorithm (Figure 7b), and ground truth label (Figure 7c). The predictability score depends on how strongly an appliance event co-occurred with a particular

[7] We normalize each image to better visualize locations with lower prediction confidence.

location. As shown in Figure 7a, most appliances related to human activities have high predictability scores (e.g., kettle, hairdryer, microwave, coffee machine, etc). On the other hand, appliances that cycle in the background (e.g., heater) have very low predictability. The stove has many clusters of background events. This is because when the stove is on, it cycles between a few power levels, and the cycle durations depend on the heating levels. Interestingly, we found that stove clusters with higher power levels ("stove-big-cycle") also have high predictability scores, while others with cycling states ("stove-cycle") show low scores. This is likely because people are next to the stove more often when the heating level is high.

We can also see that without clustering using both location predictions and event vectors, it is hard to separate some of the cluster boundaries. Besides, learning to relate energy events to location data enables us to measure the distances of events in a well-defined physical space.



(a) Location predictability scores (b) Discovered clusters (c) Ground truth appliance labels

Figure 7: t-SNE visualization of the learned event vectors colored by (a) location predictability scores (b) discovered clusters, and (c) ground truth labels.

8.4 Network Implementation and Training Details

In this section, we provide implementation and training details of our neural network model. We use convolution and deconvolution layers for the energy encoder and decoder. Each module has 8 layers with a kernel size of 3 and a stride of 2. We choose the dimensions of $z_{t,cat}$ and $z_{t,cont}$ to be 128 and 3. The location predictors have 5 layers of 3D deconvolution with a kernel size of 3 and a stride of 2 in each dimension. The frames of location images for each time window have 32 $\times$ 32 $\times$ 32 pixels. We discretize the x, y, and time dimensions into 32 points, where the range of the x-y dimensions are 10 meters and the time dimension has 32 seconds. The neural networks are implemented in Tensorflow (Abadi et al., 2016). For training, we use the Adam (Kingma & Ba, 2014) optimizer with a learning rate of 0.001 and a batch size of 64.

8.5 Clustering Parameters

In all experiments, we set $\eta_{D_{\text{loc}}} = 0.4$ meters, $\eta_z = 0.03$, $\eta_s = 0.2$, and $N_{\text{min}} = 10$. These values are chosen based on physical and computational constraints. The value of $\eta_{D_{\text{loc}}}$ is based on the minimum physical separation between two appliances. The value of η_z only affects the search space in each iteration, and is chosen to be small for computational efficiency. The minimum predictability score η_s is chosen based on a validation set from one of the homes. The value of N_{min} is set to 10 to say that we need the appliance to appear in the data at least 10 times before we trust that it is a real appliance.

8.6 Limitations

We discuss the limitations of our approach in this section. One limitation is that some remotely activated appliances may not have predictable locations. However, from our experience collecting the dataset, the vast majority of the appliances used on a daily basis (Table 2) require human interaction. For example, a person has to put food into a microwave before turning it on, to hold a hair dryer while drying hair, and to push a button to get a coffee machine running. Even for an appliance with a remote controller, as long as the person has a regular place to interact with the appliance from (e.g., always turning the TV on while sitting on the couch), our model can still learn to predict the location of interaction. Another limitation is that our location sensor has a limited coverage area (around 40 feet in radius). This is enough to cover a typical one-bedroom apartment. For a larger house, one could deploy a second sensor, similarly to how a WiFi repeater extends the coverage area.